Chair of Communication Networks
Department of Electrical and Computer Engineering
Technical University of Munich

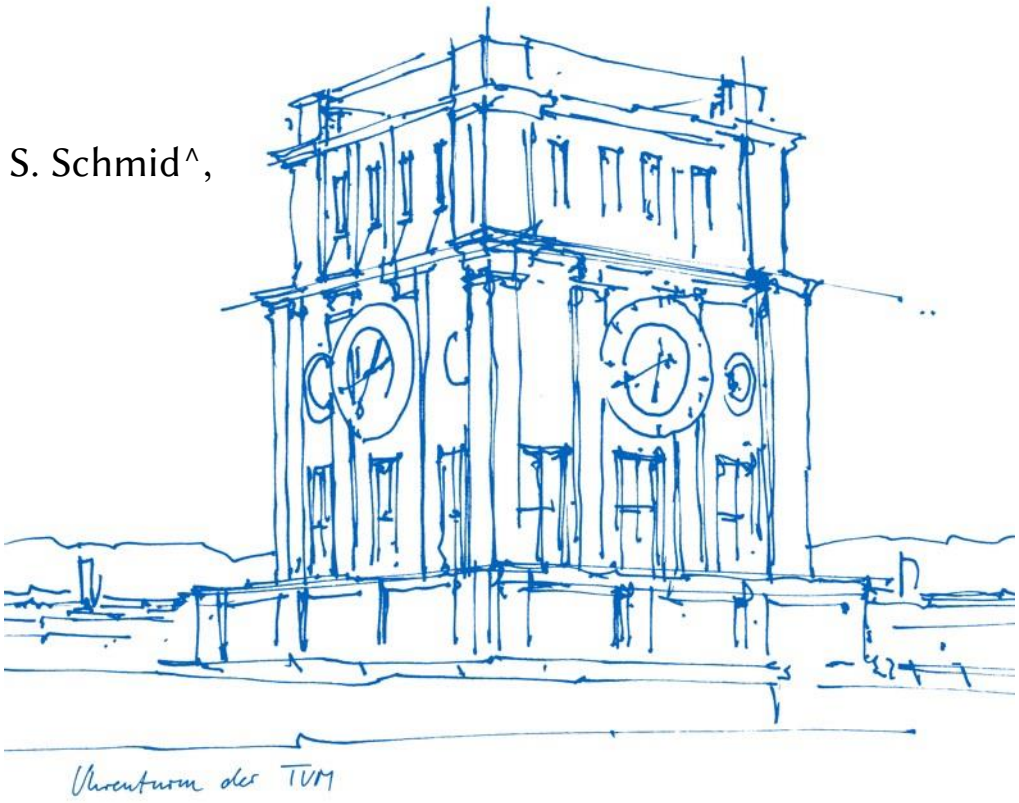
Chameleon: Predictable Latency and High Utilization with Queue-Aware and Adaptive Source Routing

A. Van Bemten*, **N. Đerić***, A. Varasteh*, S. Schmid^,
C. Mas Machuca*, A. Blenk*, W. Kellerer*

Contact: nemanja.deric@tum.de

* Technical University of Munich

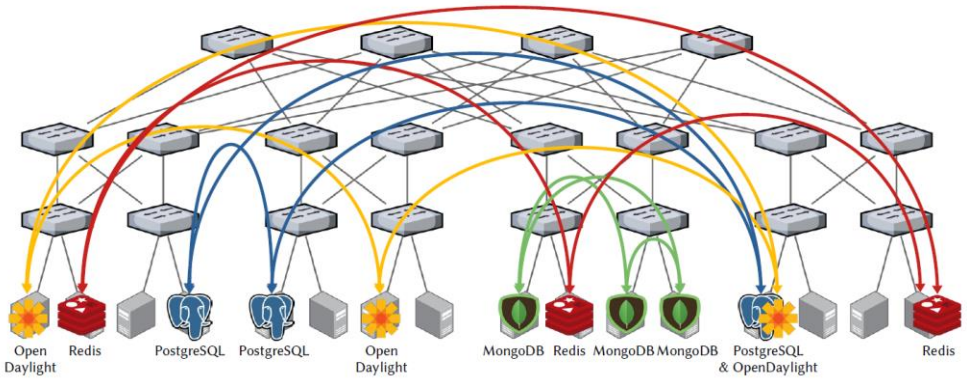
^ University of Vienna



©2020 Technical University of Munich

Chameleon: Predictable Latency and High Utilization with Queue-Aware and Adaptive Source Routing

We want to provide **strict per-packet latency guarantees**



Data-center networks

Databases and controllers synchronize for *fault-tolerance* and/or *availability*

delay violation or packet loss

↓

synchronization takes longer

↓

longer response time

↓

SLAs violations

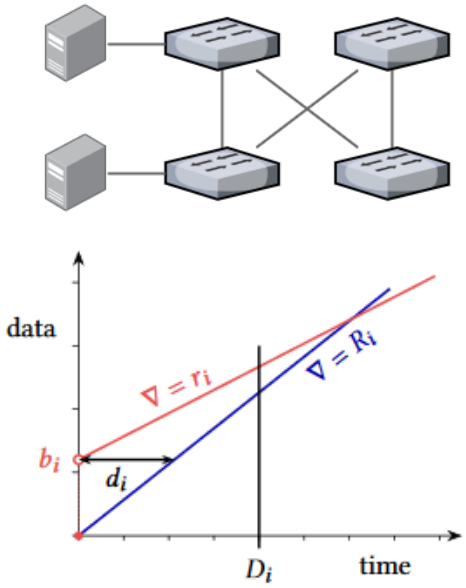
Chameleon: Predictable Latency and High Utilization with Queue-Aware and Adaptive Source Routing

Shortcomings of SotA → Unexploited optimization opportunities available in current networks

1. Sub-optimal Resource Usage
2. Static Flow Embedding

These drawbacks could lead to **unnecessarily low utilization**

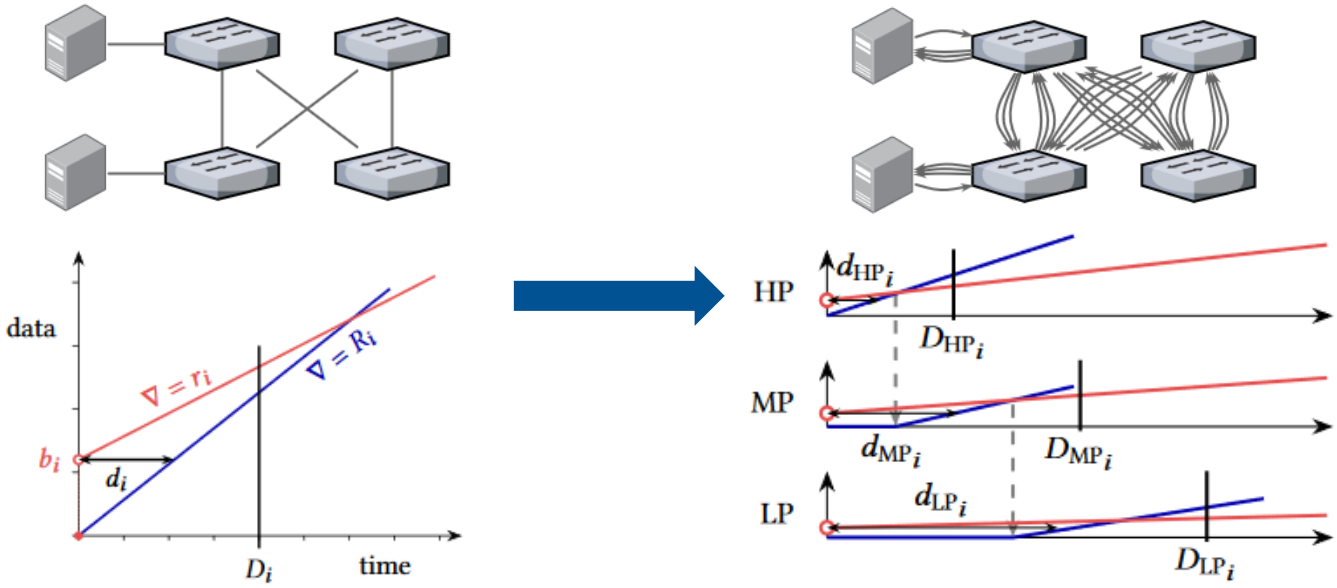
Most SoA (Silo [**SIGCOMM15**]) do not exploit optimally "advanced" switch features such as priority queueing



(a) Silo: per link.

All outgoing packets on one port will be served by the same queue → Resource & Demand Oblivious

Most SoA (Silo [**SIGCOMM15**]) do not exploit optimally "advanced" switch features such as priority queueing.



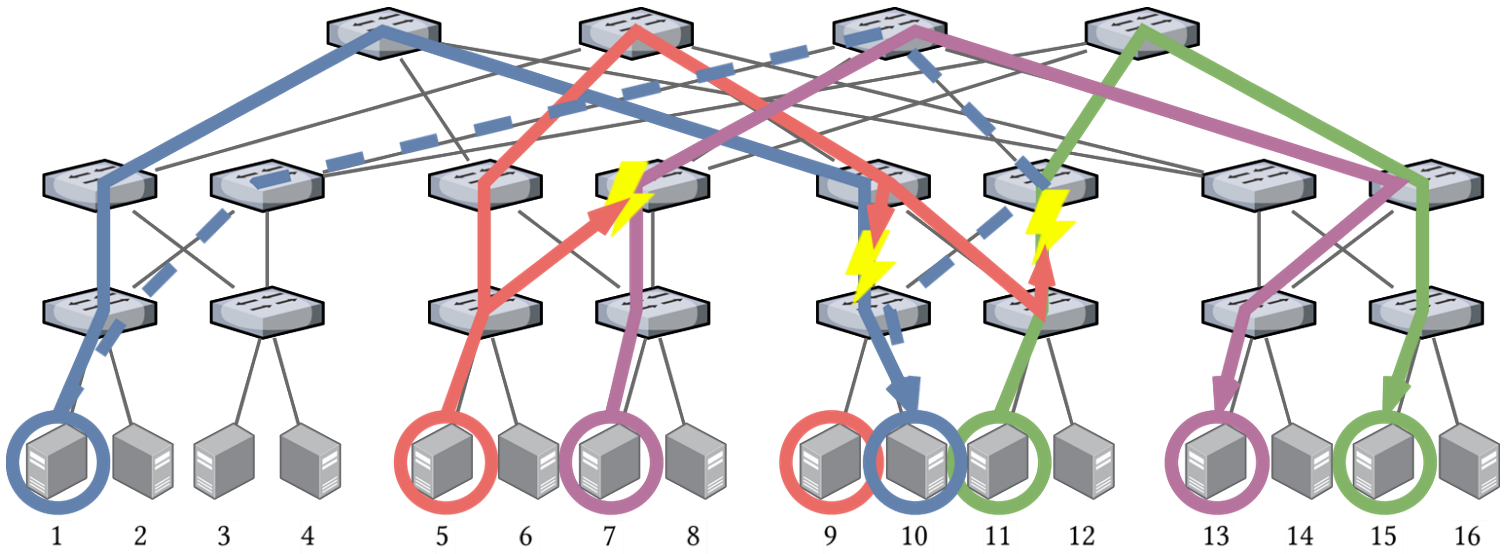
(a) Silo: per link.

(b) *Chameleon*: per queue.

Per-Queue Topology: 1) More embedding opportunities & 2) Higher offered delay diversity
Chameleon is Queue-Aware → Higher Utilization

Second problem of the SoA (Silo **[SIGCOMM15]**, QJump **[NSDI15]**) is that it is inflexible and static.

Once a decision is taken, it is **never reevaluated**

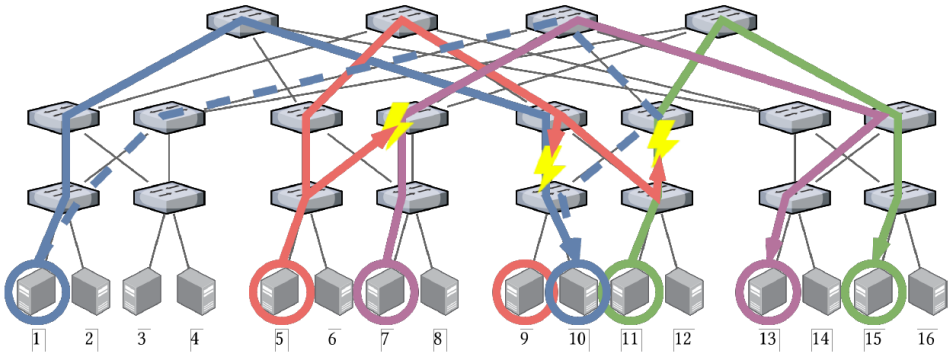


Flow/Network reconfigurations have the potential to greatly **increase:**

the number of **accepted flows** ⟶ network **utilization** & operator **revenue**

Can we actually exploit **network reconfiguration**?

That requires **reconfiguration** of switches that are forwarding delay-sensitive traffic



Can reconfigurations happen properly without interfering with the data plane performance?

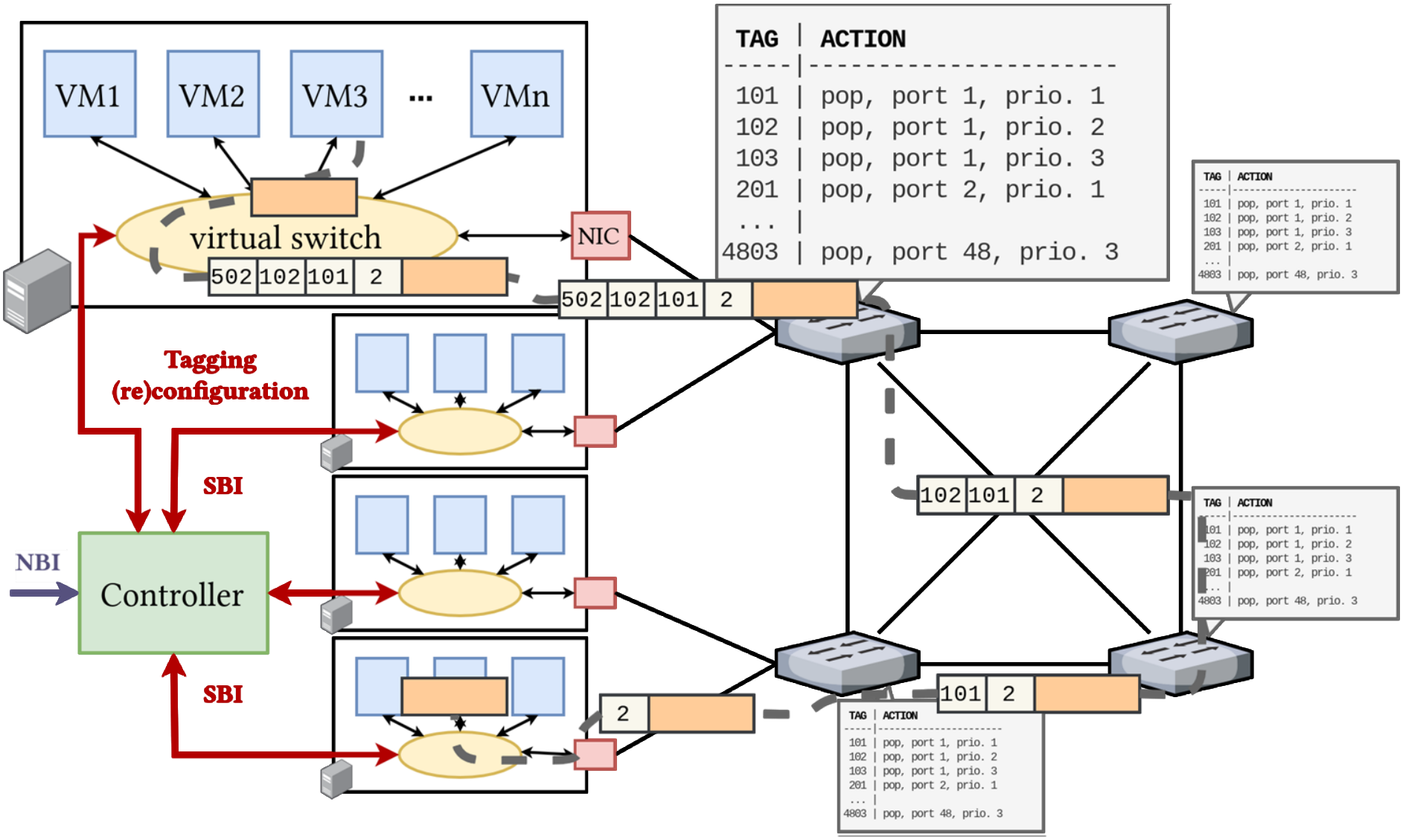
We investigated many different switches
(HP E3800, HP2920, Dell S3048-ON, Dell S4048-ON,
Pica8 P3290, Pica8 P3297, NEC PF5240),
and the answer is (unfortunately)

No!

[ANCS19]

Use Source Routing → Reconfigure the hosts instead of the switches to avoid their unpredictability
(that also circumvents the problem of consistent network updates)

Chameleon


VM1
VM2
VM3
...
VMn
virtual switch
502 102 101 2
NIC
Tagging
(re)configuration
SBI
NBI
Controller
SBI
2
102 101 2
101 2
TAG | ACTION
101 | pop, port 1, prio. 1
102 | pop, port 1, prio. 2
103 | pop, port 1, prio. 3
201 | pop, port 2, prio. 1
4803 | pop, port 48, prio. 3

Flow Embedding Strategy

When the centralized controller receives a flow request, the following algorithm is run

```
1:  function EMBEDDINGSTRATEGY(request)
2:      response ← ROUTE(request)
3:      if response ≠ NULL then
4:          RESERVE(response), return response
5:      for each flowToReroute in LIM(SORT(GETFLOWSTOREROUTE(request))) do
6:          INCREASEGRAPHCOSTS(flowToReroute, request)
7:          reroutingResponse ← ROUTE(flowToReroute)
8:          if reroutingResponse ≠ NULL then
9:              RESERVE(reroutingResponse)
10:             FREE(flowToReroute.originalPath)
11:             response ← ROUTE(request)
12:             if response ≠ NULL then
13:                 RESERVE(response), return response
14:     return NULL
```

Firstly, we use per-queue level
k calculus
embed a

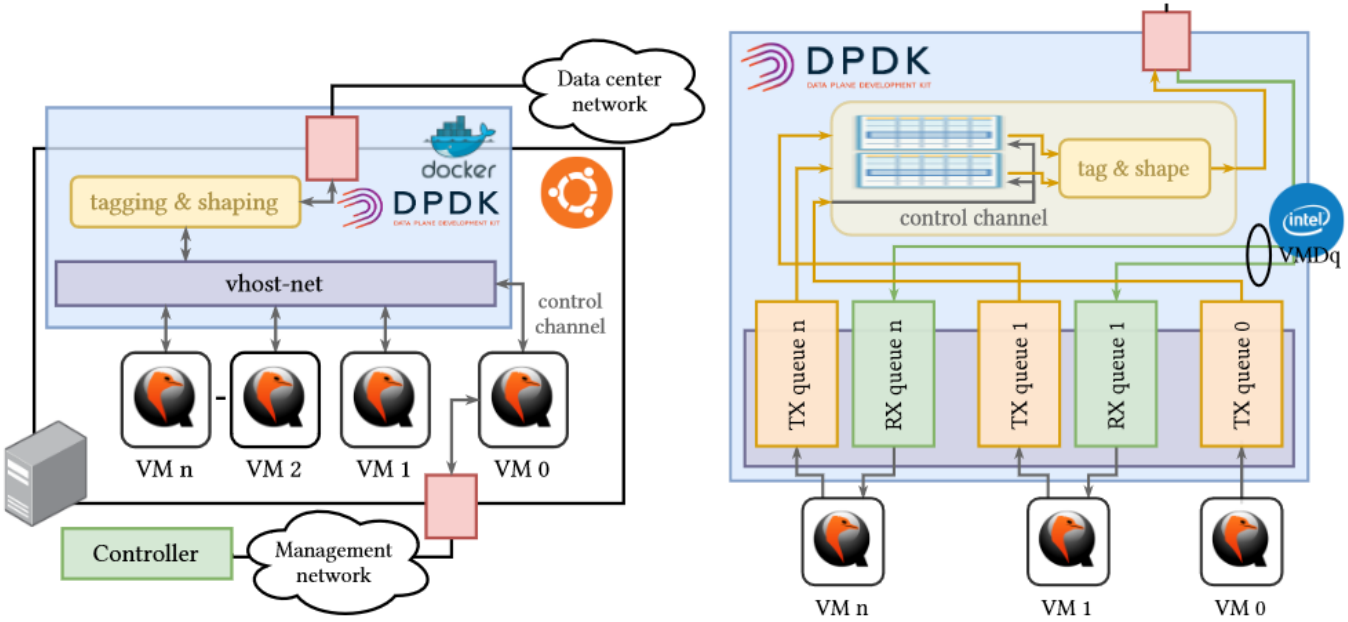
If a flow can be embedded, we simply reserve resources and configure the corresponding end host.

If a flow can't be embedded, we run a rerouting algorithm.

Chameleon

End-Host Implementation

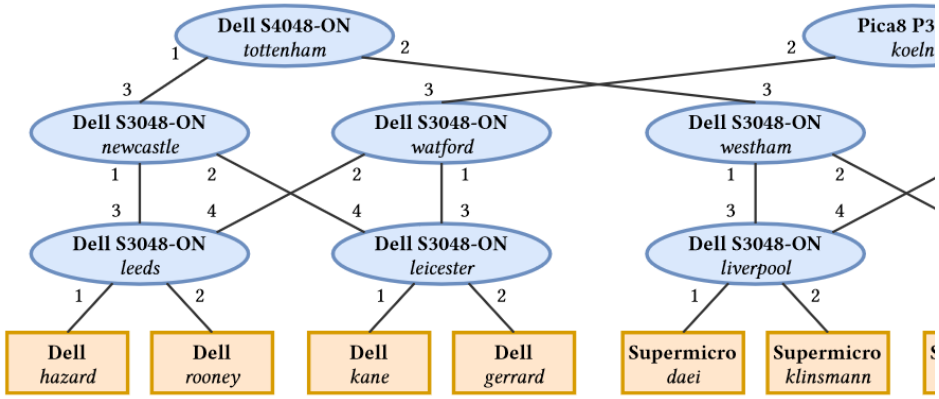
1. DPDK App (VMDq based) with tagging and shaping (leaky token bucket)
2. Vagrant software to deploy VMs
3. vhost-net/virtio-net architecture to interconnect DPDK app and VMs



(a) Server. (b) DPDK application.

Evaluations – Testbed Measurements & Simulations

1. Fat tree topology with k = 4 (or more in simulations)
2. Various flow types, listed in Table
    1. Flows are added in an online fashion until the first embedding failure
3. Live Monitoring is done with EndaceDAG measurement card

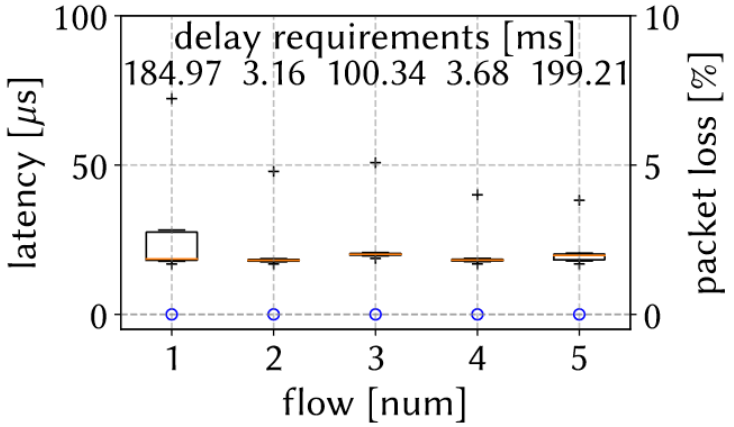


Flow description	Rate	Burst	Deadline
Category 1: Industrial applications (IA) [1, 34]			
Database operations	[300, 550] Kbps	[100, 400] byte	[80, 120] ms
SCADA operations	[150, 550] Kbps	[100, 400] byte	[150, 200] ms
Production control	[100, 500] Kbps	[100, 400] byte	[10, 20] ms
Control and NTP	[1, 100] Kbps	[80, 120] byte	[10, 20] ms
Category 2: Clock synchronization (CS) [51]			
PTP	[1, 220] Kbps	[80, 300] byte	[2, 4] ms
Category 3: Control plane synchronization (CPS) [2, 55]			
Eventual consistency	[2, 4] Mbps	[80, 140] byte	[50, 200] ms
Strict consistency	[5, 8] Mbps	[1000, 3000] byte	[50, 200] ms
Adaptive consistency	[2, 4] Mbps	[80, 120] byte	[50, 200] ms
Category 4: Bandwidth-hungry applications (BH) [4, 5, 45, 65]			
Hadoop, data-mining	[100, 150] Mbps	[1000, 5000] byte	[10, 100] ms
Hadoop, data-mining	[100, 200] Mbps	[1000, 3000] byte	[10, 100] ms
Hadoop, data-mining	[80, 200] Mbps	[1000, 3000] byte	[50, 100] ms

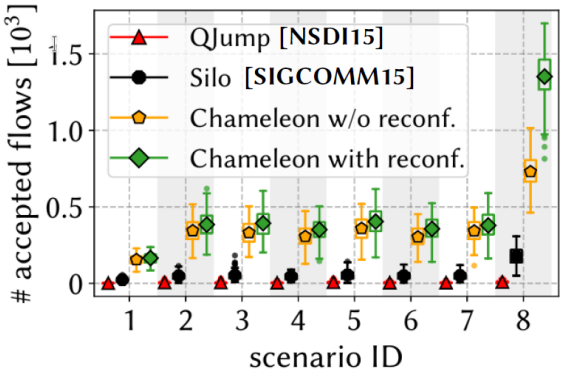
Table 1: Considered flow types and their characteristics.

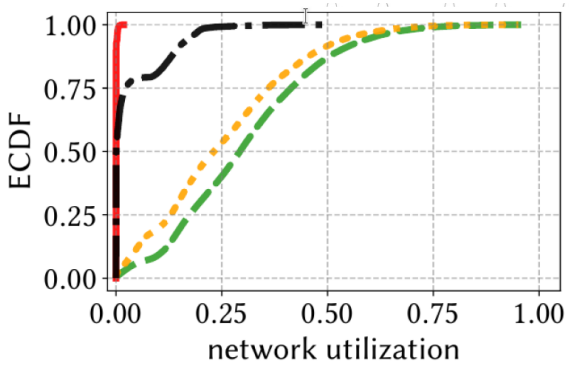
Evaluations – Results

Does Chameleon actually work? Yes



Does Chameleon bring any benefits? Yes, up to 15 times more flows compared to SoA.





resource-aware and reconfigurable networks can **improve cloud network utilization** while providing predictable latency

Thank you for your time!

For more details check the paper:

Chameleon: Predictable Latency and High Utilization with Queue-Aware and Adaptive Source Routing